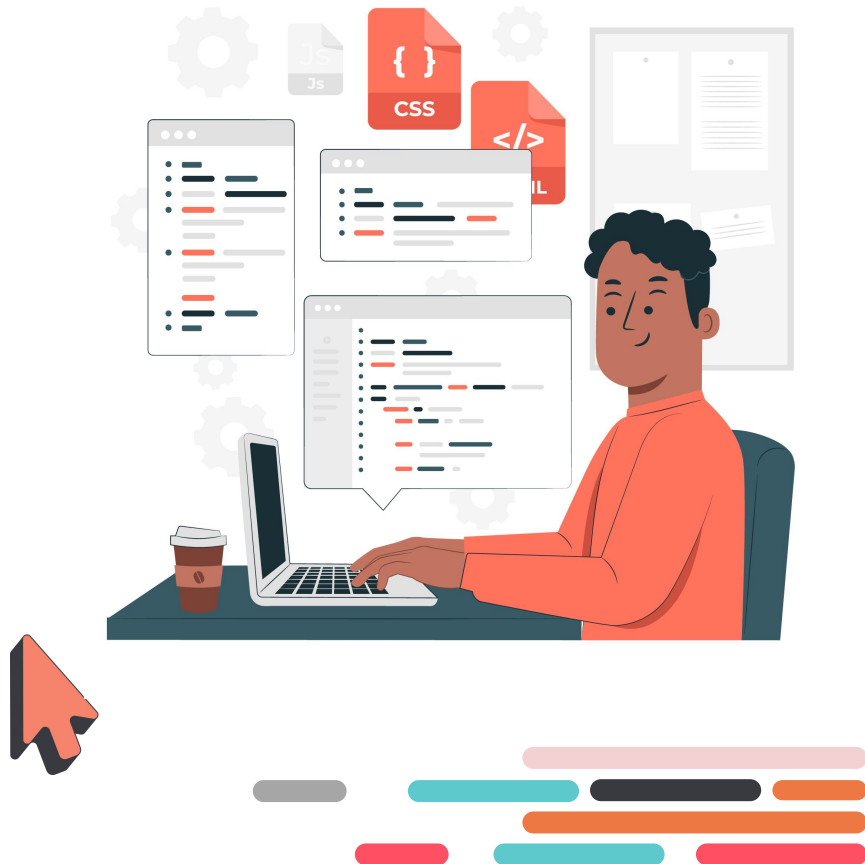




KEMNAKER



HTML, CSS & Dasar Responsive Website





KEMNAKER

sanbercode



Outline

- HTML
- CSS
- Dasar Responsive Website





HTML Dasar





KEMNAKER



Pengenalan HTML

HTML merupakan singkatan dari Hyper Text Markup Language. HTML adalah sebuah bahasa standar untuk pembuatan halaman web. Dengan adanya HTML, kita dapat membedakan struktur yang tersusun dari sebuah halaman melalui tag atau elemen-elemen penyusunnya.

Elemen atau tag pada HTML dikenali oleh browser seperti google chrome, firefox atau Ms Edge, dll. Browser tersebut mengidentifikasi setiap elemen penyusun HTML dan ditampilkan sesuai karakteristik elemen tersebut.





KEMNAKER



Struktur HTML

berikut ini
struktur dasar HTML:

```
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
</html>
```



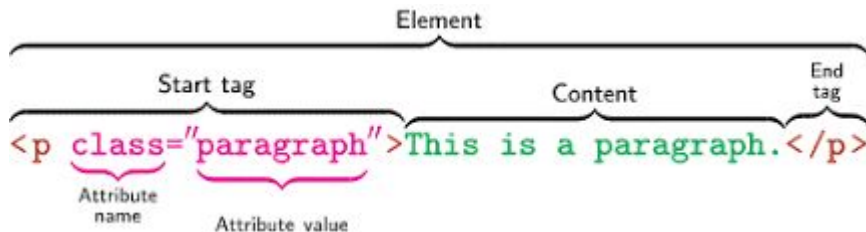


KEMNAKER



Element HTML

Elemen atau sering disebut dengan tag pada HTML adalah penyusun dari sebuah halaman web yang dapat dikenali oleh browser dan ditampilkan oleh browser sesuai karakteristik masing-masing. Sebuah elemen HTML terdiri dari tag pembuka(start/open tag), atribut (attribute), isi (content), dan tag penutup (end tag). Berikut ini ilustrasi Tag pada html



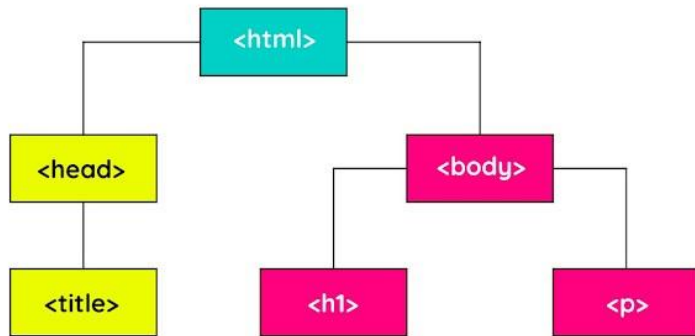


KEMNAKER



Hirarki HTML

Didalam Sebuah elemen HTML dapat terisi dengan elemen-elemen lain di dalamnya. Semua dokumen HTML yang kita lihat di halaman web terdiri dari kumpulan elemen bersarang atau elemen yang mengandung elemen lain di dalamnya. Contohnya setelah tag pembuka dan sebelum penutup sebuah elemen, kita dapat meletakkan sebuah konten. Konten dapat berupa teks ataupun sebuah Tag HTML yang lain.





Element Dasar HTML

berikut ini adalah beberapa element dasar pada HTML:

- heading (h1, h2, dst.)
- p
- div
- img
- a
- ul, ol, li
- form





CSS Dasar





Pengenalan CSS

Cascading Style Sheet atau dikenal dengan CSS merupakan bahasa style sheet yang berguna untuk membantu menyajikan dokumen yang ditulis dengan HTML. CSS dipakai untuk mendesain halaman depan atau tampilan website (front end). Menggunakan CSS kita bisa mengatur warna , ukuran , posisi , background, dan lain lain. Sehingga ada nilai estetika yang ditambahkan dalam tampilan suatu website.





Selector , Property dan Value

Pada CSS ada beberapa istilah biasa kita temukan diantaranya : selector , property, dan value. Kita mempelajari terlebih dahulu pengertian dari ketiga istilah tersebut.

selector Property Value

a { Background-color : Yellow ; }

Declaration start Declaration End





Cara Penulisan Kode CSS dalam HTML

untuk penulisan kode css atau penerapan kode css pada HTML terdapat 3 cara yaitu:

- Internal CSS
- External CSS
- Inline CSS





Internal CSS

```
<!DOCTYPE html>
<html>
<head>
  <style type="text/css">
    p{
      font-family: Helvetica;
      line-height: 2.00 em;
      font-size: 20px;
      color: orange;
    }
    h2 {
      font-family: sans;
      color: rgb(10, 8, 8);
    }
  </style>
</head>
<body>
  <h2>Hello CSS</h2>
  <p>Hello aku bisa CSS</p>
</body>
</html>
```





External CSS

```
<!DOCTYPE html>
<html>
<head>
  <link href="style.css" rel="stylesheet" >
</head>

<body>

  <h2>Hello CSS</h2>
  <p>Hello aku bisa CSS</p>
</body>
</html>
```

```
p{
  font-family: Helvetica;
  line-height: 2.00 em;
  font-size: 20px;
  color: orange;
}

h2 {
  font-family: sans;
  color: rgb(10, 8, 8);
}
```



Inline CSS

```
<body>

  <h2 style=" font-family: Helvetica;
    line-height: 2.00 em;
    font-size: 20px;
    color: ■orange;">
    Hello CSS</h2>

  <p style = "font-family: Helvetica;
    line-height: 2.00 em;
    font-size: 20px;
    color: ■orange;">
    Hello aku bisa CSS</p>
</body>
```



Property Dasar CSS

berikut ini adalah beberapa property dasar pada HTML:

- **color**
- **font-family**
- **font-size**
- **text-align**
- **text-decoration**
- **width**
- **height**





Semantic HTML





KEMNAKER



Semantic HTML

Apa yang dimaksud dengan semantic tag? Semantic Tag atau Semantic Markup, adalah sebutan untuk tag-tag HTML yang memiliki 'arti' atau 'makna'. Kata semantic berasal dari kata yunani yang berarti 'mempelajari arti' (the study of meaning).

HTML ditujukan untuk membuat struktur halaman web. Tag `<p>` misalnya, digunakan untuk membuat paragraf, tag `<h1>` untuk membuat judul/header, dan tag `<table>` untuk membuat tabel, kesemua tag ini memiliki 'arti' yang jelas, karena itu termasuk kedalam semantic tag.





Contoh Semantic Tag

Beberapa semantic tag yang dibawa oleh HTML5 adalah sebagai berikut:

- `<article>` — untuk membuat elemen artikel;
 - `<figcaption>` — untuk membuat teks caption pada figure;
 - `<figure>` — untuk membuat figur atau gambar pada artikel;
 - `<footer>` — untuk membuat elemen bagian kaki dari web;
 - `<header>` — untuk mebuat kepala kop dari web;
 - `<nav>` — untuk membuat navigasi;
 - `<section>` — untuk membuat bagian artikel;
 - `<time>` — untuk membuat elemen yang menyatakan waktu;
- dan masih banyak lagi.

contoh lainnya bisa dicek melalui <https://learntheweb.courses/topics/html-semantics-cheat-sheet/>





KEMNAKER



Mengapa Harus Pakai Elemen Semantik?

Salah satu keuntungan menggunakan elemen semantik adalah dokumen HTML kita akan mudah dibaca, baik itu oleh manusia maupun mesin.

elemen semantik juga bagus untuk SEO. Jadi kalau mau websitemu disukai mesin pencari, sebaiknya gunakanlah elemen ini.





KEMNAKER



Contoh Kode HTML tanpa Semantic Tag

```
<div id="header"></div>
<div class="section">
  <div class="article">
    <div class="figure">
      <img>
      <div
class="figcaption"></div>
    </div>
  </div>
</div>
<div id="footer"></div>
```





KEMNAKER



Contoh Kode HTML dengan Semantic Tag

```
<header></header>
<section>
  <article>
    <figure>
      <img>
      <figcaption></figcaption>
    </figure>
  </article>
</section>
<footer></footer>
```





Table HTML





KEMNAKER



Table HTML

Table di dalam HTML dibentuk dengan sebuah baris (row), tepatnya menggunakan element `<tr>` yang merupakan kepanjangan dari table row. Adapun element pendukung lain yang membentuk sebuah table adalah `<th>` (table header) dan `<td>` (table data). `<td>` menunjukkan sebuah cell, sedangkan `<th>` menunjukkan cell induk dan ditandai dengan tulisan cetak tebal.

Element lain yang berkaitan dan digunakan dalam sebuah HTML table adalah `<caption>`, `<col>`, `<colgroup>`, `<thead>`, `<tfoot>` dan `<tbody>`.





KEMNAKER



Contoh kode table

Contoh Lainnya Dapat dilihat melalui https://www.w3schools.com/html/html_tables.asp

```
<table>
  <thead>
    <tr>
      <th>No.</th>
      <th>Nama</th>
      <th>Tempat Lahir</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1.</td>
      <td>John</td>
      <td>London</td>
    </tr>
  </tbody>
</table>
```





Validasi Form HTML





Validasi Form HTML

Perihal membuat form di HTML, bukan hanya tentang bagaimana cara untuk mendapatkan data yang dimasukkan oleh pengguna. Beberapa hal harus dipertimbangkan untuk membuat form agar dapat digunakan dengan baik dan memenuhi tujuan awal

Salah satunya, validasi data form. Validasi data form diperlukan untuk memastikan bahwa data input oleh pengguna telah sesuai dengan setiap tujuan elemen form

Misalnya, kita membuat field input untuk menerima data nama, maka pengguna harus memasukkan data nama mereka (bukan data nomor telepon, dll)





KEMNAKER



Beberapa Contoh Validasi Form HTML

berikut ini beberapa contoh validasi form HTML:

- type number
- required
- min dan max
- maxLength dan minLength
- type email
- type password
- dan lain-lain

contoh lainnya bisa cek di https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation





KEMNAKER



Sesi Tanya Jawab





Box Model





Box Model

Box model adalah salah satu properti dasar yang harus kita pahami sebelum mempelajari hal lain dari CSS. Pada setiap elemen pasti memiliki properti ini. Jika kita ibaratakan sebuah elemen HTML merupakan suatu barang maka box model adalah kardus pembungkus dari barang itu. Didalam kardus tidak hanya terdiri dari barang itu saja.

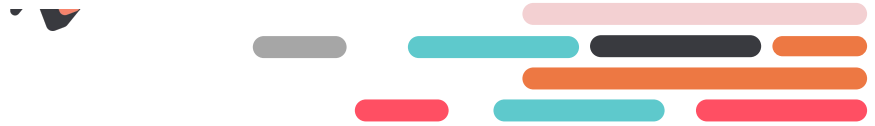
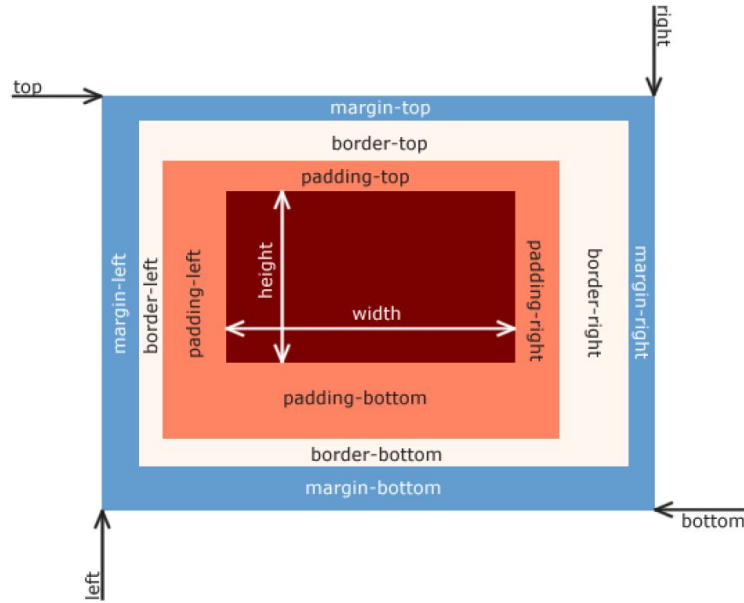
Cakupan Box Model diantaranya:

1. Padding
2. Border
3. Margin





Ilustrasi Box Model





CSS Position





KEMNAKER



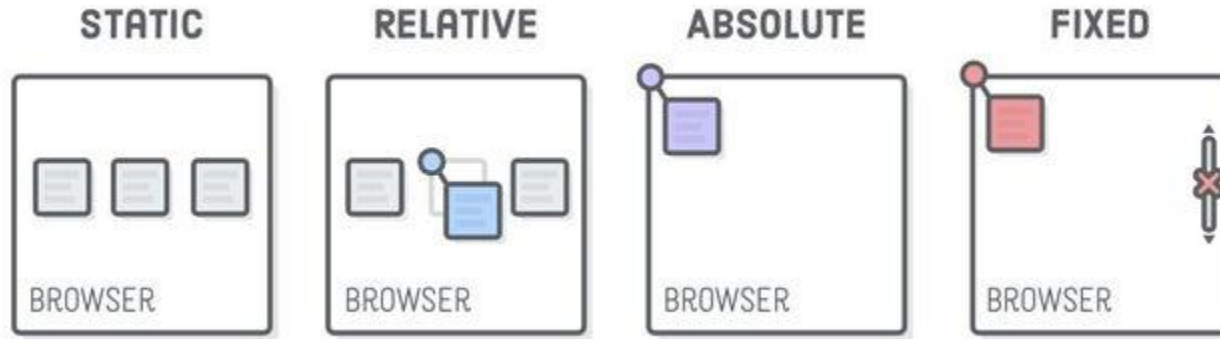
Apa itu CSS Position?

Position merupakan property css yang digunakan untuk menentukan sifat posisi suatu elemen di suatu document. Sifat di sini dalam artian elemen bisa berubah, tetap , dan lain lain . beberapa value dari property position diantaranya:

1. **Static** : Value static merupakan nilai default dari position suatu element. Value static akan membuat suatu element bertumpukan.
2. **Relative** : Value relative digunakan untuk membuat suatu element bisa dipindah pindahkan, kita bisa memindahkan elemen tersebut dengan property seperti top , left. Sehingga elemen tersebut akan menempa layernya elemen lain.
3. **Absolute** : Value absolute digunakan untuk membuat suatu elemen bisa dipindahkan dan membuat elemen mengambang ke atas dibanding element lain. Bedanya dengan property relative, pada property absolute layer bawahnya akan bergeser seolah olah berada di bawah layer dengan property relative.
4. **Fixed** : Value fixed digunakan untuk membuat elemen mengambang di antara elemen elemen lain dan juga bisa dipindahkan posisinya. Property ini mirip dengan property absolute. Bedanya dengan property absolute adalah property dengan value fixed posisinya akan tetap berada di posisinya terhadap layar meskin layar page scroll.



Ilustrasi CSS Position





CSS Display (Block dan Inline)





KEMNAKER



Memahami CSS Display (Block dan Inline)

Display adalah salah satu property dalam CSS yang berguna untuk mengatur "tampilnya" suatu elemen. Display memiliki beberapa value, dengan default value inline. Namun, pada masing-masing elemen tag pada html memiliki default value yang berbeda, default value yang paling sering digunakan adalah inline dan block.





Display Inline

Elemen yang memiliki property ***display: inline;*** hanya memblok sesuai dengan konten yang ada pada elemen tersebut. Kita **tidak dapat mengubah atau mengatur nilai lebar dan tinggi** dari elemen tersebut. Selain itu, nilai **margin-top dan margin-bottom juga tidak berpengaruh** pada elemen inline.

Contoh elemen html yang memiliki **default value inline**, yaitu:

- ``
- `<a>`
- ``





Display Block

Elemen yang memiliki property `display: block`; selalu membuat blok baru pada tampilan. Elemen akan membentang dari kiri hingga ke kanan mengisi penuh lebar dari halaman browser. Kita dapat melakukan manipulasi banyak hal pada elemen yang bernilai `block`. Baik mengatur lebar, tinggi, margin sesuai dengan kebutuhan yang kita inginkan.

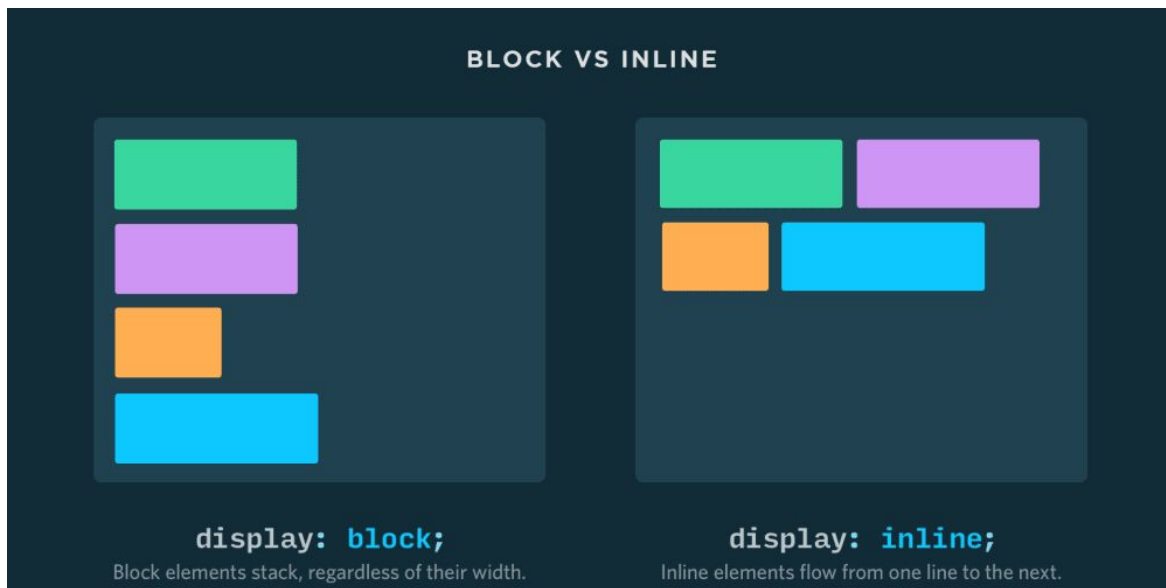
Contoh **default value block**, yaitu :

- `<div>`
- `<p>`
- `<form>`
- `<header>`
- `<footer>`
- `<section>`





Ilustrasi Display Inline dan Block





Flexbox





KEMNAKER



Belajar menggunakan flexbox

flexible Box biasa disebut flexbox merupakan mode layout yang ada di CSS3 dan digunakan untuk mengatur elemen di suatu halaman web. Flexbox ini akan mengatur ukuran dari elemen anaknya secara otomatis, dan mampu beradaptasi dengan ukuran container-nya.

Tujuan dari flexbox yaitu memberikan container kemampuan untuk mengatur panjang, lebar, dan posisi item-item yang berada di dalamnya agar memaksimalkan ruang yang ada. Pengaturan ini sangat penting bagi seorang frontend developer untuk membuat sebuah website yang nyaman dilihat di berbagai device dengan berbagai macam resolusi.





Cara Mendefinisikan flexbox

kode HTML:

```
<nav class="container">
  <div>Home</div>
  <div>Search</div>
  <div>Logout</div>
</nav>
```

kode CSS:

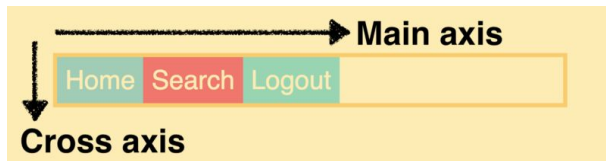
```
*{
  border: 1px solid red;
}

.container {
  display: flex;
}
```

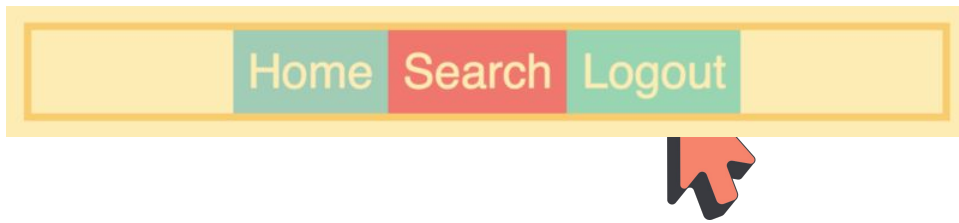


hubungan display : flex dengan Justify Content dan Align Items

Justify-content dan align-items adalah dua properti CSS yang membantu kita mendistribusikan item-item di dalam container. Mereka mengontrol bagaimana item diposisikan secara horizontal (main axis) maupun vertikal (cross axis).



Mari kita buat ketengah ketiga item secara horizontal menggunakan “justify-content: center;” akan menghasilkan:





KEMNAKER



cheatsheet justify content untuk display flex

`justify-content: flex-start;`



`justify-content: flex-end;`



`justify-content: center;`



`justify-content: space-between;`



`justify-content: space-around;`



`justify-content: space-evenly;`



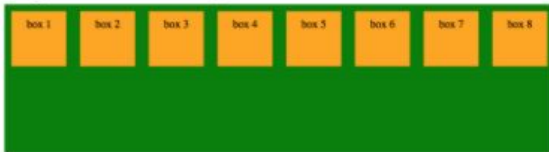


KEMNAKER

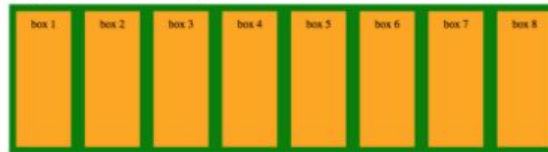


cheatsheet align-items untuk display flex

`align-items: flex-start;`



`align-items: stretch;`



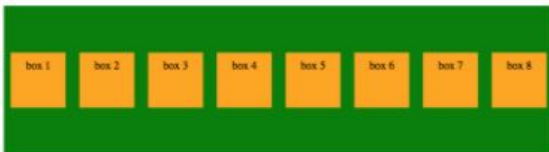
`align-items: flex-end;`



`align-items: baseline;`



`align-items: center;`





Pseudo Class





KEMNAKER



Apa itu Pseudo Class?

Pseudo-class dalam CSS digunakan untuk memberikan gaya pada elemen berdasarkan keadaan atau kondisi tertentu. Pseudo-class ditandai dengan menggunakan tanda titik dua setelah nama elemen, diikuti oleh nama pseudo-class. Berikut adalah beberapa contoh pseudo-class umum dalam CSS:

1. `:hover`
2. `:focus`
3. `:active`
4. `:nth-child()`
5. `:first-child` dan `:last-child`
6. `:not`
7. dan lain-lain

untuk contoh lainnya bisa cek di https://www.w3schools.com/css/css_pseudo_classes.asp





Animations





Bagaimana cara implementasi Animation pada CSS ?

pada CSS3 kita dapat membuat animation tanpa menggunakan javascript ataupun flash. Yang terpenting untuk kita membuat animation di css adalah keyframes.

Apa itu keyframes? keyframes adalah suatu alur animasi pada CSS3 untuk merubah 1 style ke style yang lain.





KEMNAKER



Implementasi Keyframes

```
/* The animation code */
@keyframes example {
  from {
    background-color: red;
  }

  to {
    background-color: yellow;
  }
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

```
/* The animation code */
@keyframes example {
  from {
    background-color: red;
  }

  to {
    background-color: yellow;
  }
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```



KEMNAKER



Contoh lainnya dengan menggunakan keyframes

```
/* The animation code */
@keyframes example {
  0% {
    background-color: red;
  }

  25% {
    background-color: yellow;
  }

  50% {
    background-color: blue;
  }

  100% {
    background-color: green;
  }
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```





KEMNAKER



Sesi Tanya Jawab





Pengertian Responsive





Apa itu Responsive ?

Responsive Web Design merupakan kemampuan design web agar bisa disesuaikan berdasarkan ukuran browser yang digunakan. Responsive Web Design tidak sama dengan mobile design. Mobile design perlu menciptakan sebuah situs atau web app yang sama sekali baru dengan konten khusus yang diciptakan untuk mobile experience.

Responsive web design menjadi dasar dari penyebaran sebuah situs web, menggunakan fleksibel atau fluid grid (grid yang dapat berubah-ubah), fluid images (gambar yang fleksibel atau dapat berubah-ubah), dan CSS3 Media Queries agar bisa beradaptasi dengan pengguna yang menggunakan perangkat dengan lebar dan resolusi berbeda.





Bagaimana cara melihat responsive website di browser?

Dengan cara klik kanan di browsernya > inspect, maka tampilannya akan berbentuk mobile (ini merupakan dev tools dari browser chrome)

Dan juga jangan lupa mengatur viewport pada tag html didalam <head>

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```





Relative Units





Relative Units

Inti dari Responsive adalah Relative CSS Units. Ini adalah units yang mendapatkan nilainya dari beberapa nilai external lainnya. Ini berguna karena memungkinkan, misalnya, lebar gambar harus berdasarkan pada lebar browser yang akan ditampilkan.

relative units yang paling umum adalah:

- %
- em
- rem
- vw
- vh





Media Queries





CSS media queries

CSS media queries digunakan untuk membatasi ruang pengesekusian kode CSS, artinya CSS yang kita buat melalui media queries ini hanya berjalan di ukuran lebar layar tertentu,

syntaxnya :

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

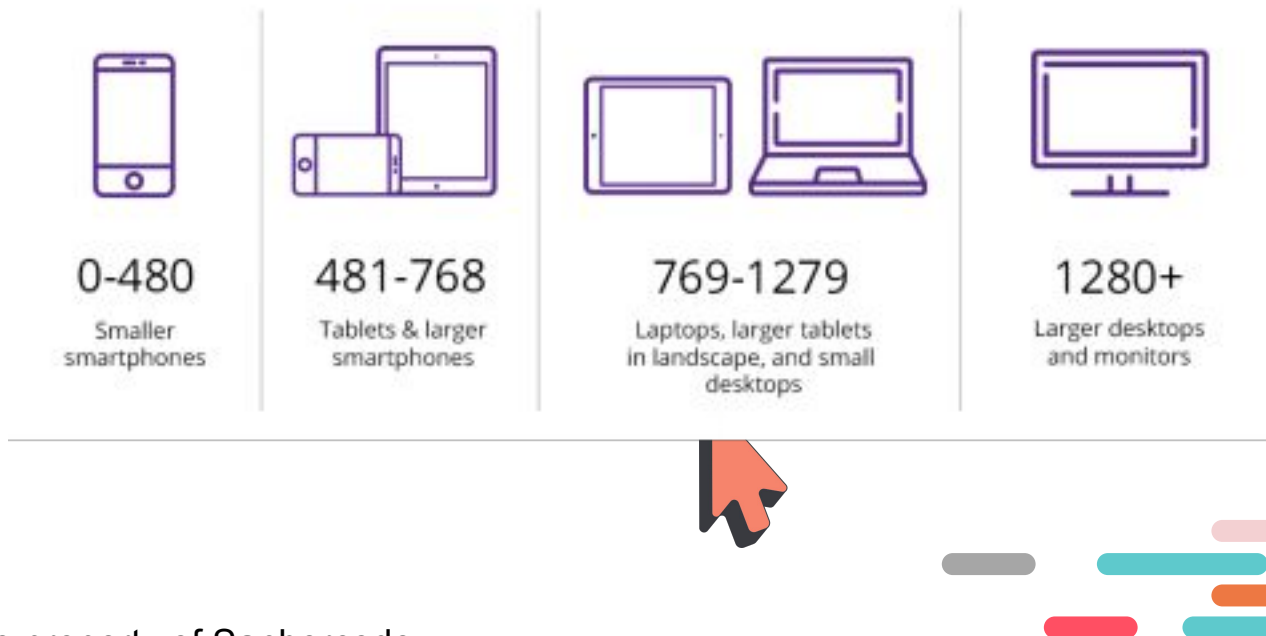




KEMNAKER



Patokan Ukuran Layar berdasarkan Devicenya





Display Grid





Dasar CSS Grid

CSS Grid merupakan salah satu fitur baru di CSS 3 yang fungsi utamanya adalah mengatur layout pada website. Meskipun penggunaannya belum sepenuhnya kompatibel di semua browser, namun CSS Grid diprediksi akan menjadi salah satu fitur penting di masa depan untuk mengatur layout halaman pada HTML responsif.

CSS Grid merupakan cara baru dalam menyusun layout menggunakan CSS, sebelumnya orang banyak menggunakan float. Lihat informasi CSS Grid pada caniuse.com (<https://caniuse.com/#feat=cssgrid>) untuk mengecek kompatibilitas dengan browser yang digunakan.



Mendefinisikan CSS Grid

Untuk mendefinisikan css grid, kita akan menggunakan property css display :

display : grid

Disini hanya akan ada beberapa contoh penerapan grid yang akan kita coba :

1. Mendefinisikan Grid
2. Membuat Layout Kolom Grid
3. Dan mengatur jarak





KEMNAKER



Mendefinisikan Grid

kode HTML:

```
<div class="container">
  <div class="box">1</div><div
class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div><div
class="box">5</div>
</div>
```

kode CSS:

```
.container{
  display: grid;
}
.box{
  border: 2px solid black; min-height: 100px; width: 100%; font-size: 50px;
}
```





KEMNAKER



Membuat Layout Kolom Grid

kode HTML Tetap sama

kode CSS:

```
.container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}

.box{
  border: 2px solid black;
  min-height: 100px;
  width: 100%;
  font-size: 50px;
}
```





KEMNAKER



Mengatur jarak antar kolom maupun baris

kode HTML Tetap sama

kode CSS:

```
.container{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  column-gap: 50px;
  row-gap: 50px;
}
.box{
  border: 2px solid black;
  min-height: 100px;
  width: 100%;
  font-size: 50px;
}
```





KEMNAKER

sanbercode



Kesimpulan

- HTML
- CSS
- Dasar Responsive Website





Terima Kasih

